

LA-UR-14-24814

Approved for public release; distribution is unlimited.

Title: Gromacs IC Tutorial

Author(s): Junghans, Christoph
Phillips, Joshua L.
Wall, Michael E.

Intended for: Report

Issued: 2014-07-03 (rev.1)

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

GROMACS IC Tutorial

Christoph Junghans
(incl. slides from Joshua Phillips and Mike Wall)

Los Alamos National Laboratory
NM, USA

June 27, 2014

Plan for the Day

- Installing GROMACS
- From PDB to MD
- Lunch
- HPC Tips & Tricks
- Question & Answer Session
- Advanced Simulations
 - Spectral clustering (MDSCTK)
 - Coarse-graining (www.votca.org)
- What is next ?

Installing GROMACS Dependencies

GROMACS is a GPL code and free of charge.

Hard Dependencies

- C compiler
 - cmake
 - GNU make (or MSVS or Google Ninja)

Soft Dependencies

- fftw (can be built-in, legal implications)
 - Intel MKL
 - Nvidia Cuda

Installing GROMACS

Considerations

Single vs. Double precision

- Why do you want double?
- Can you afford a factor two in speed?
- Do you do NVE or normal mode analysis?

MPI vs. Threads

- Are you going to use more than one node (shared memory)?
- Communication overhead vs. performance gain

Intel vs. GNU

- Are you an Intel fan boy?
- Are you using the latest gcc/icc/MKL/fftw ?

Installing GROMACS

Log-on into Darwin

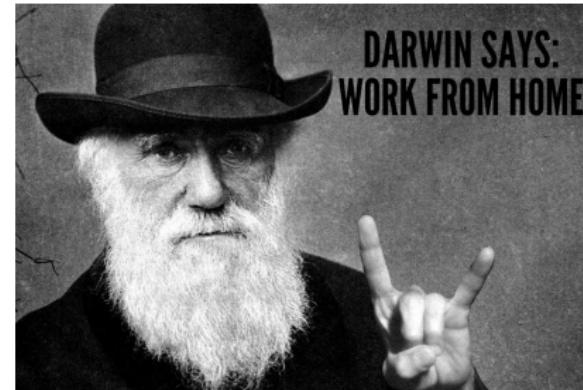
Linux/Mac Os

- Open a console
- ssh
MONIKER@darwin.lanl.gov
add -X option if possible
- Password: crypto card

Windows

- Download putty
- host: darwin.lanl.gov
- user name: MONIKER
- Password: crypto card

FNs: Remember to have a netlogin window open!



Get a node

- salloc
--reservation=gromacs
- squeue | grep \$USER
last word is the node
number (cnXXX)
- ssh <NODE>

Installing GROMACS

Preparations

- Setup proxy:
`export ftp_proxy=proxyout:8080`
- Download GROMACS: wget ↵
`ftp://ftp.gromacs.org/pub/gromacs/gromacs-4.6.5.tar.gz`

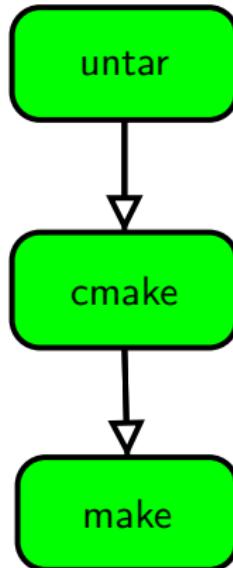
- Unpack tarball:
`tar -xzvf gromacs-4.6.5.tar.gz`

- Make a build directory:

```
cd gromacs-4.6.5  
mkdir build  
cd build
```

- Run cmake:

```
module load cmake  
cmake .. (or ccmake or cmake-gui)
```



Installing GROMACS

Cmake Options

Add options to cmake -DSOME=THING -DSOMETHING=ELSE . .

Useful options

- Enable/Disable MPI: -DGMX_MPI=<ON | OFF>
if disabled make sure to use: -DGMX_THREAD_MPI=ON
- Enable/Disable GPU: -DGMX_GPU=<ON | OFF>
- Install prefix: -DCMAKE_INSTALL_PREFIX=<PATH>
- Single or Double: -DGMX_DOUBLE=<ON | OFF>
- Enable minimal viewer: -DGMX_X11=<ON | OFF>
- Enable built-in fftw: -DGMX_BUILD_OWN_FFTW=ON

In short: cmake . . -DGMX_X11=ON ←

-DCMAKE_INSTALL_PREFIX=/panfs/scratch1/\$USER/gromacs

Installing GROMACS

Build & Install

- Get number of cores: `grep -c processor /proc/cpuinfo`
- Build: `make -j<number of cores>` (add `VERBOSE=1` for verbose output)
- Install: `make -j<number of cores> install`
- Setup paths: `source←
/panfs/scratch1/$USER/gromacs/bin/GMXRC.bash`
- Run: `g_luck`

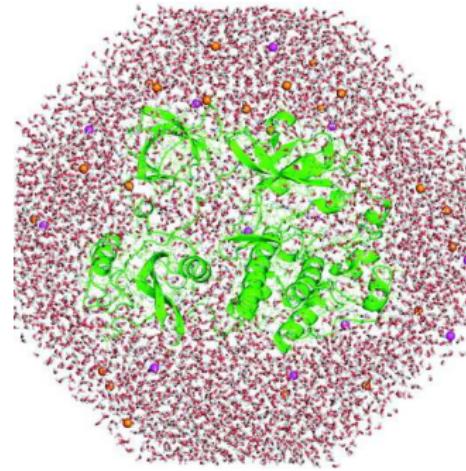
“Move about like a Scientist, lay down, get kissed”
(Red Hot Chili Peppers)

From PDB to MD

Introduction

PDB is a standard format for protein structures, which contains:

- Atom positions
- Atom types
- Residue name & number
- No topological information (bonds, angle)
- No force field information
- No solvent



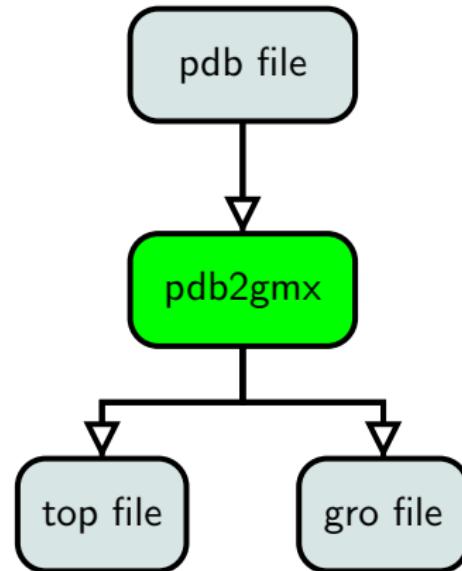
Src kinase protein, from:
M. Karplus and J.
Kuriyan, PNAS
102, 6679-6685 (2005).

From PDB to MD

Considerations

pdb2gmx Inputs

- Force field: interactive or `-ff` option
- GROMACS supports multiple force fields!
- Water model: interactive or `-water` option
- Interactive termini selection or `-ter`
- Hydrogen removal: `-ignh` option
- Hydrogen to virtual site: `-vsites` hydrogens option



From PDB to MD

Force Fields in a Nutshell

- Force fields describe the interactions between the atoms
- Incl. their functional form
- Bonded and non-bonded interactions

Non-bonded interactions

$$U(r) = 4\epsilon \left[(\sigma/r)^{12} - (\sigma/r)^6 \right]$$

or

$$U(r) = C_{12}/r^{12} - C_6/r^6$$

or Buckingham or tabulated

Bonded interactions

- Bonds
- Angle
- Dihedrals

Supported Force Fields

- CHARMM
- Amber
- GROMOS
- OPLS
- Martini

Forcefield Family

“Philosophy”

GROMOS

Match free enthalpies of hydration and apolar solvation for a wide range of compounds.
Relative polar/apolar free enthalpies are relevant for many biomolecular processes (eg. folding, association, membrane formation/transport).

AMBER

Match first principles, QM calculations and measurements from NMR.
“General FF” with parameters for nucleic acids, general organic compounds.

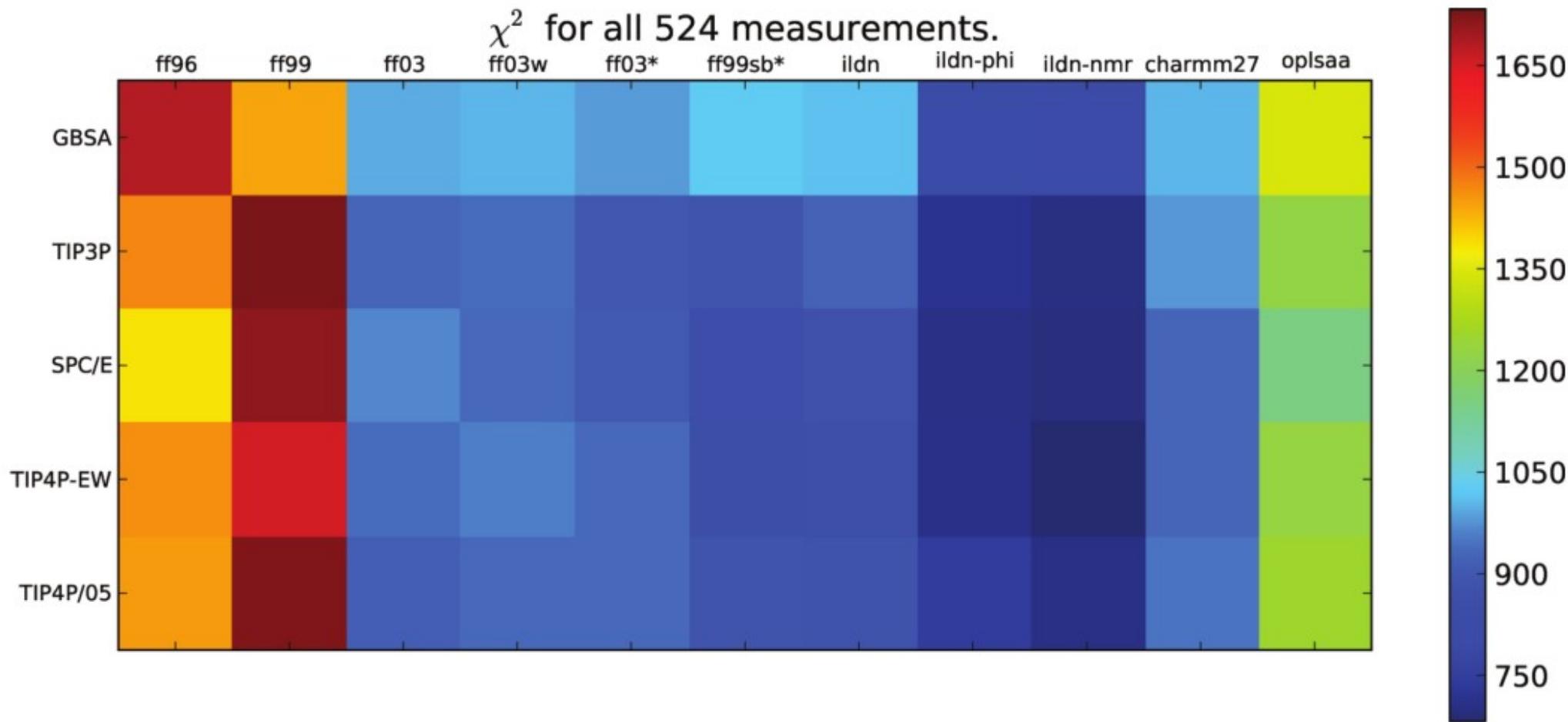
CHARMM

Similar to AMBER, but with slightly different fitting methods to the QM calculation data.

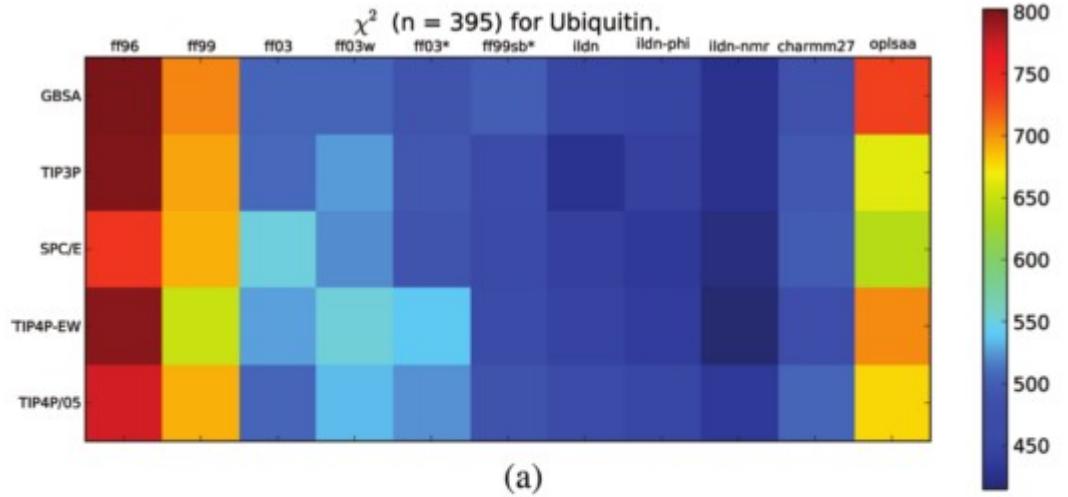
OPLS/AA

More broad acceptance of compatible water models, but similar to CHARMM in many respects.

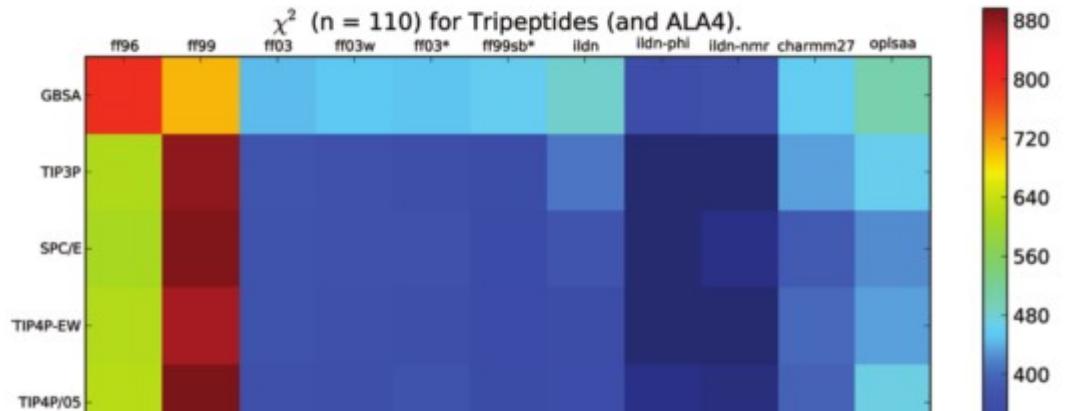
χ^2 for all 524 measurements.



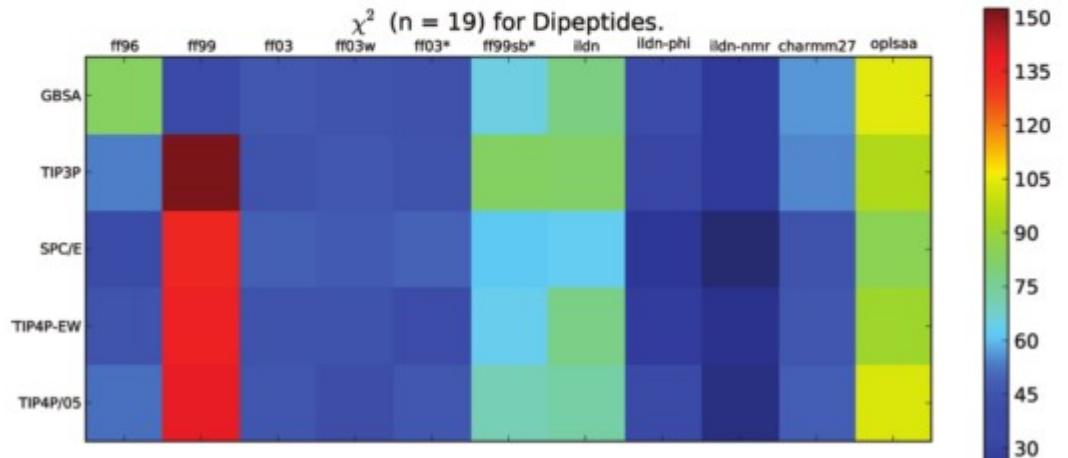
Beauchamp, K. a, Lin, Y.-S., Das, R., & Pande, V. S. (2012). Are Protein Force Fields Getting Better? A Systematic Benchmark on 524 Diverse NMR Measurements. *Journal of Chemical Theory and Computation*, 8(4), 1409–1414. doi:10.1021/ct2007814



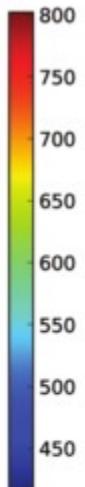
(a)



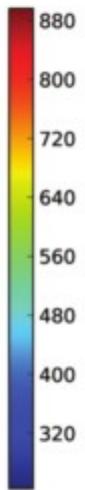
(b)



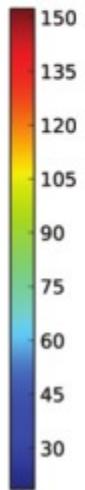
(c)

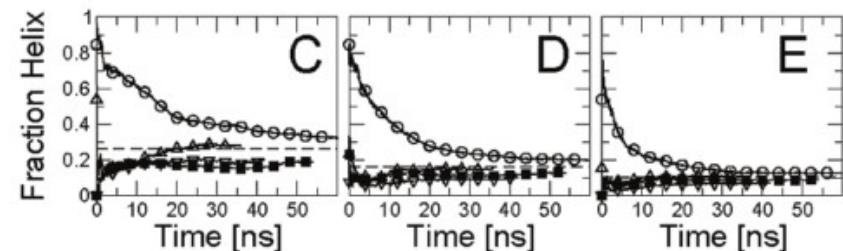
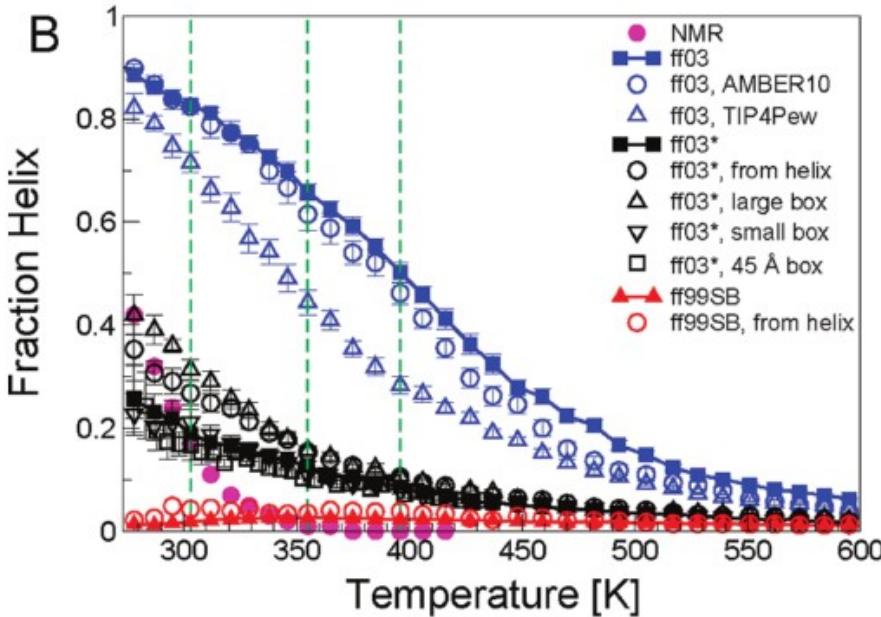
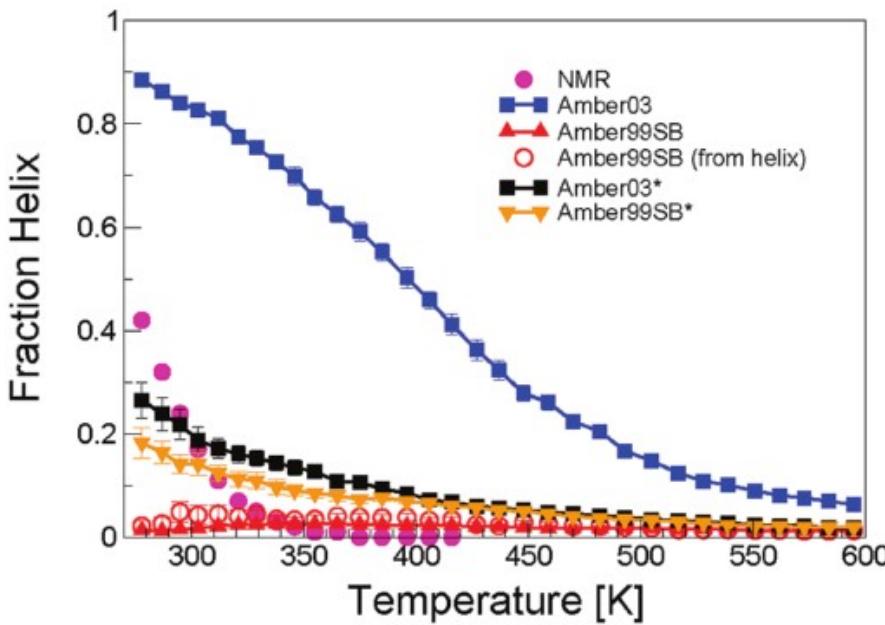


Forcefields vary in performance, seemingly making constant (perhaps irreconcilable) tradeoffs when matching different types of experimental data.



The jury is still out, but the current consensus for proteins is that recent CHARMM or AMBER variants are a good choice.



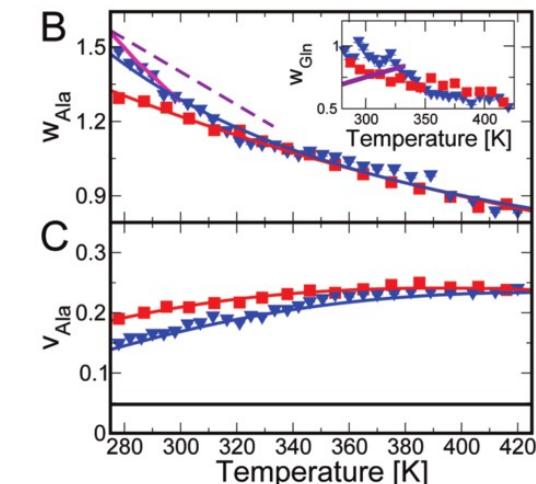
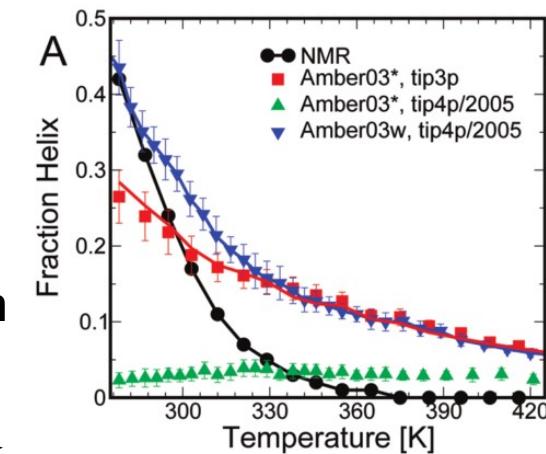


Best, R. B., & Hummer, G. (2009). Optimized molecular dynamics force fields applied to the helix-coil transition of polypeptides. *The Journal of Physical Chemistry. B*, 113(26), 9004–15. doi:10.1021/jp901540t

The parameters are updated to match more general properties specific to proteins as well, such as the percentage of helix content.

The parameters don't live in a bubble, the water/ion/solvent model parameters can greatly impact results as well.

Best, R. B., & Mittal, J. (2010). Protein simulations with an optimized water model: cooperative helix formation and temperature-Induced unfolded state collapse. *The Journal of Physical Chemistry B*, 114(46), 14916–14923. doi:10.1021/jp108618d

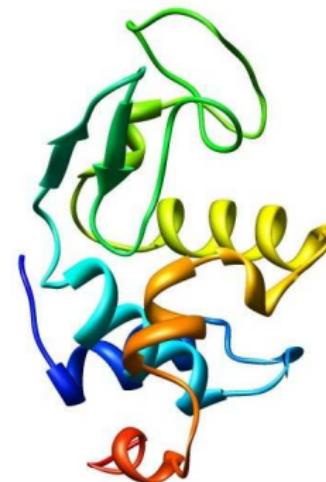


From PDB to MD

1AKI Conversion

This tutorial is modeled after Justin Lemkul's 1AKI tutorial¹

- Goto <http://www.rcsb.org/>
- Get 1AKI as PDB (text)
- Copy to darwin
 - scp 1AKI.pdb←
MONIKER@darwin:.
- pdb2gmx -f 1AKI.pdb
- Select OPLS-AA force field and SPC/E water
- Look at topol.top

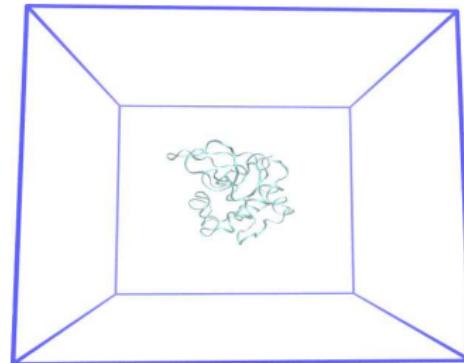


¹Justin's tutorial([link](#))

From PDB to MD

1AKI Solvating

- Put 1AKI into a cubic box with distance 1 nm from the boundary
`editconf -f conf.gro -o conf2.gro -c -d 1.0 -bt cubic`
- Solvate in spce water
`genbox -cp conf2.gro -cs spc216.gro -o conf3.gro -p topol.top`
- Compare topol.top with #topol.top.1# (backup file)
- Create runtime setting file



From PDB to MD

Runtime Settings

Create grompp.mdp:

integrator	= steep
emtol	= 1000.0
emstep	= 0.01
nsteps	= 50000
nstlist	= 1
rlist	= 1.0
coulombtype	= PME
rcoulomb	= 1.0
rvdw	= 1.0
pbc	= xyz

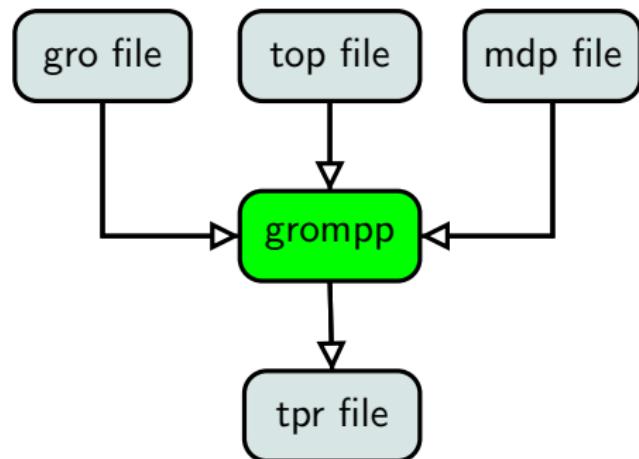
- Use steepest descent (not really an integrator)
- Initial step size 0.01nm
- Minimize until max force is 1000.0 kJ/mol/nm
- All cutoffs = 1nm
- Use smooth PME for electrostatic
- Use periodic boundary conditions in xyz
- Need more details? See the gromacs manual!

From PDB to MD

GROMACS Preprocessor

- Compile a tpr file
`grompp -c conf3.gro`
- Uh, a charged system
- To check defaults look at
`mdoutmdp`
- Add counter ions
`genion -o conf4.gro ←`
`-p topol.top -pname←`
`NA -nname CL -nn 8`
- Select SOL to replace Water molecules
- Compare `topol.top` with
`#topol.top.2#` (backup file)

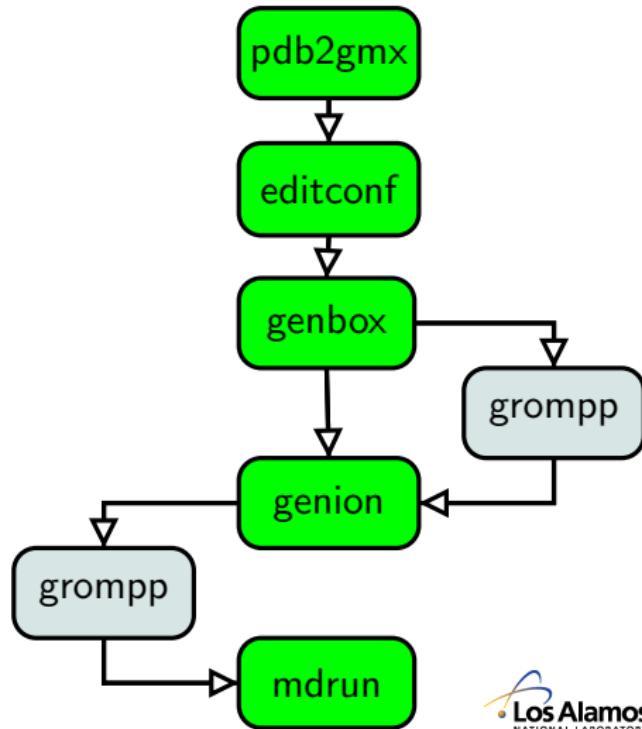
Note: tpr is actually the only file needed for the simulation (except tables)



From PDB to MD

1AKI Minimization

- Recompile tpr
`grompp -c conf4.gro`
- Minimize `mdrun -v`
- Look at `md.log`
- Calculate energy (not actually calculated, but just converted) `g_energy`
- Select potential energy
- Look at `energy.xvg` using `xmgrace`



MD in a Nutshell

GROMACS stands for "GROningen Mixture of Alchemy and Childrens' Stories"² and is a molecular dynamics code optimized for bio-simulations.

$$m_i \ddot{\vec{r}_i} = \sum_j F_{ij}(r_{ij}) = \dot{\vec{p}_i}$$

Molecular dynamics is the “art” of integrating Newton’s equation of motion.

²Or: "GROningen MAchine for Chemical Simulation"

MD in a Nutshell

Integrators

Simple idea: Taylor expansion

Velocity Verlet Algorithm

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \frac{\Delta t}{m_i} \vec{p}_i(t) + \frac{\Delta t^2}{2m_i} \vec{F}_i(t) + \mathcal{O}(\Delta t^3),$$

$$\vec{p}_i(t + \Delta t) = \vec{p}_i(t) + \frac{\Delta t}{2} \left(\vec{F}_i(t) + \vec{F}_i(t + \Delta t) \right) + \mathcal{O}(\Delta t^3)$$

- Only correct up to 2nd order
- One force call per time step
- Higher order will need more force calculations per time step

MD in a Nutshell

Integrators

Alternative Formulation:

Leap-Frog Algorithm

$$\vec{p}_i(t + \Delta t/2) = \vec{p}_i(t - \Delta t/2) + \Delta t \vec{F}_i(t) + \mathcal{O}(\Delta t^2)$$

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \frac{\Delta t}{m_i} \vec{p}_i(t + \Delta t/2) + \mathcal{O}(\Delta t^3),$$

- Leap-Frog and Velocity Verlet differ when combined with Thermostats and Barostats
- What is a good integrator? A symplectic integrator!³

³See the gromacs manual!

MD in a Nutshell

Ensembles

- Newton's equation conserved energy → NVE
integrator=md
- Thermostat → NVT
tcoupl=...
- Barostat → NPT
pcoupl=...

Ensemble should correspond to experimental conditions!

MD in a Nutshell

Thermostats

Thermostats are additions to equations of motion to go from NVE to NVT.

4 Groups:

- Local thermostats couple to every particle
- Global thermostats couple to the total kinetic energy
- Stochastic thermostats have random variables
- Deterministic thermostats have no random variables

MD in a Nutshell

Rescaling Thermostats

- Simple rescaling thermostat (Global, deterministic)
 - Rescales total kinetic energy $K' = \lambda K$ with $\lambda = \sqrt{T/T_0}$
 - Iso-kinetic ensemble
- Berendsen thermostat (Global, deterministic)⁴
`tcouple=Berendsen`
 - Rescales with a finite rate:

$$\frac{dK}{dt} = \frac{K(T) - K(T_0)}{\tau}$$

- “flying ice cube” problem
- No canonical ensemble!

⁴Berendsen et al., JCP **81**, 3684 (1984)

MD in a Nutshell

Rescaling Thermostats

- Stochastic velocity rescaling thermostat (global)⁵
`tcouple=v-rescale`
 - Chooses kinetic energy from the canonical distribution:
$$\frac{dK}{dt} = \frac{K(T) - K(T_0)}{\tau} + 2\sqrt{\frac{K(T)K(T_0)}{\tau}}\eta$$
 - Canonical ensemble!
 - equivalent to Langevin thermostat in K
 - “flying ice cube” problem
 - Localization: acts on kinetic energy of separate groups independently

⁵Bussi, Donadio, Parrinello JCP **126**, 014101 (2007)

MD in a Nutshell

Stochastic Dynamics

- Add a stochastic and damping force to each particle:

$$m_i \ddot{\vec{r}_i} = \vec{F}_i - \xi_i \vec{v}_i + \sigma_i \vec{\eta}_i ,$$

with $\xi_i k_B T = \sigma_i^2$ (Fluctuation-Dissipation theorem)

- Every particle is coupled to its own distribution
- Changes the dynamics
- Stabilizes dynamics → bigger time step
- Also known as Langevin thermostat
- No momentum conservation
- Can be used as implicit solvent
- `integrator=sd`

MD in a Nutshell

Andersen Thermostats

- Choose velocities randomly from a Maxwell distribution⁶
- Used to generate starting velocities
`gen_vel=yes`

⁶ Andersen, JCP **72**,2384 (1980)

MD in a Nutshell

Nosé-Hoover Thermostats

- Add an extra degree of freedom to the Hamiltonian

$$H = \sum_{i=1}^N \frac{\vec{p}_i}{2m_i} + U(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) + \frac{p_\xi^2}{2Q} + N_f kT\xi$$

- Deterministic
- Virtual mass Q needs to be chosen
- Extension: Nosé-Hoover chain - thermostat the thermostat⁷.
nh-chain-length=X

$$H = \sum_{i=1}^N \frac{\vec{p}_i}{2m_i} + U(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) + \sum_{k=1}^M \frac{p_{\xi_k}^2}{2Q'_k} + N_f kT\xi_1 + kT \sum_{k=2}^M \xi_k$$

- tcouple=nose-hoover

⁷Martyna, Klein and Tuckerman, JCP**97**, 2635 (1992)

MD in a Nutshell

Barostats are extensions to the equations of motion to go from NVT to NPT.

- Berendsen barostat
`pcoupl=Berendsen`

- Rescales simulation box to the right pressure with a constant rate

$$\frac{\partial \mathbf{P}}{\partial t} = \frac{\mathbf{P}_0 - \mathbf{P}}{\tau_p}.$$

- Parrinello-Rahman barostat⁸
`pcoupl=Parrinello-Rahman`
 - Adds degree of freedom for the box vectors

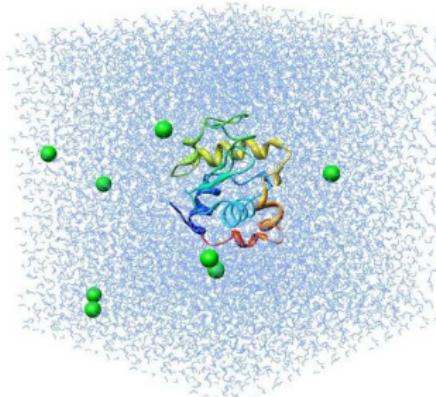
$$\frac{\partial \vec{b}^2}{\partial t^2} = V \mathbf{W}^{-1} \vec{b}'^{-1} (\mathbf{P} - \mathbf{P}_{ref}).$$

⁸Parrinello and Rahman, J Appl Phys **52**, 7182 (1981)

MD Simulations

1AKI constrained NVT equilibration

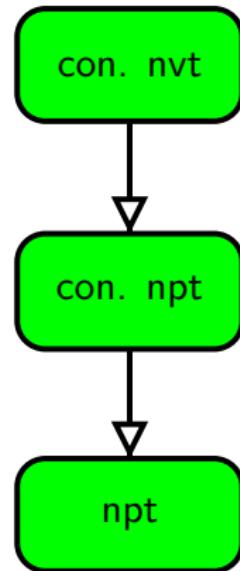
- Use new directory:
`mkdir nvt; cd nvt`
- Copy over confout.gro
`cp ../confout.gro conf.gro`
- Copy over topol.top and posre.itp
- Get Christoph's mdp file
`cp ~junghans/nvtmdp← grompp.mdp`
- Let's discuss it.
- Compile a tpr file: grompp
- Run mdrun -v
- Use g_energy to look at the pressure



MD Simulations

1AKI constrained NPT Equilibration

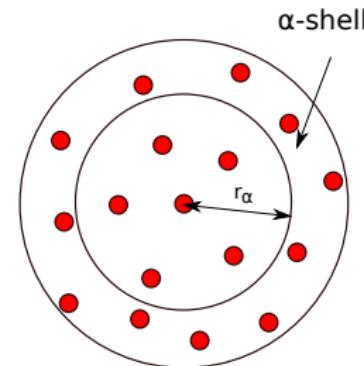
- Use new directory:
`mkdir npt; cd npt`
- Copy over confout.gro
`cp ../nvt/confout.gro conf.gro`
- Copy over topol.top and posre.itp
- Get Christoph's mdp file
`cp ~jung hans/npt.mdp grompp.mdp`
- Let's discuss the input
- Compile a tpr file: grompp
- Run mdrun -v
- Use g_energy to look at the box size



MD Simulations

1AKI MD Production

- Use new directory:
`mkdir md; cd md`
- Copy over confout.gro
`cp ../npt/confout.gro conf.gro`
- Copy over topol.top
- Get Christoph's mdp file
`cp ~junguhans/md.mdp grompp.mdp`
- Let's discuss the input
- Compile a tpr file: grompp
- Run mdrun -v
- Use g_energy to look at the energies
- Use g_rdf to calculate the radial dist. function

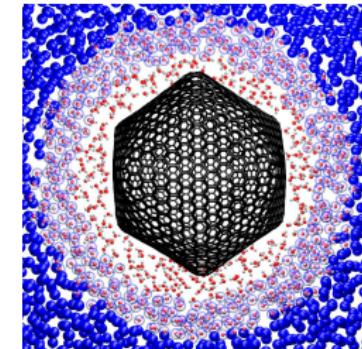


MD Simulations

Outlook

Things we have not discussed:⁹

- Free energy and PMF calculations
- Replica exchange (mpi version only)
- Coarse-grained simulations
- GROMACS supports multiple force fields
- Shell molecular dynamics simulations
- Adaptive resolution simulations
- Essential dynamics
- Implicit solvent simulations
- many, many more analysis features (try `g_-<TAB><TAB>`)
- ...



⁹See the gromacs manual!

GROMACS on HPC

Introduction

"GROMACS Runs One Microsecond At Cannonball Speeds" using:

- Optimized neighbor list for water
- Vectorized force kernels
- Parallelization:
 - Domain Decomposition
 - Threaded particle blocks (Verlet only)
 - Threaded PME calculations
 - Offload to GPUs (Verlet only)
- Separated nodes for PME calculations (use `g_tune_pme`)

Don't panic in case of a doubt let `mdrun` decide!

GROMACS on HPC

Test System

- Create topol.top:

```
#include "gromos43a1.ff/forcefield.itp"  
#include "gromos43a1.ff/spce.itp"  
[ system ]  
Water  
[ molecules ]
```

- Create a box of water

```
genbox -cs spc216.gro -p topol.top -box 5 5 5 -o  
conf.gro
```

- Get Christoph's mdp file

```
cp ~junghans/benchmark.mdp grompp.mdp
```

- Build different system sizes using: genconf -nbox X X X

GROMACS on HPC

Considerations

Options of mdrun to play

- -nb
- -nt
- -ntomp
- -ntmpi
- -dd
- -dlb
- -npme

Other tweaks

- MPI vs threads
- Group scheme
(cutoff-scheme=Group)
- Verlet scheme
(cutoff-scheme=Verlet)

GROMACS on HPC

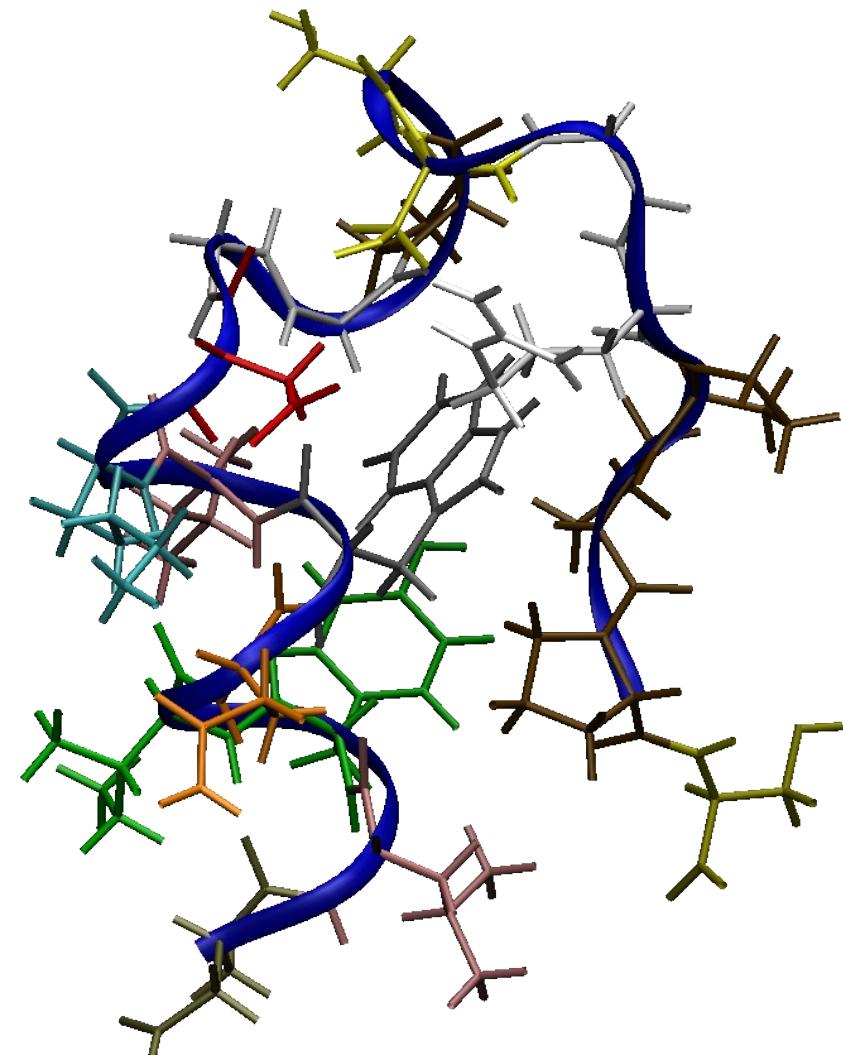
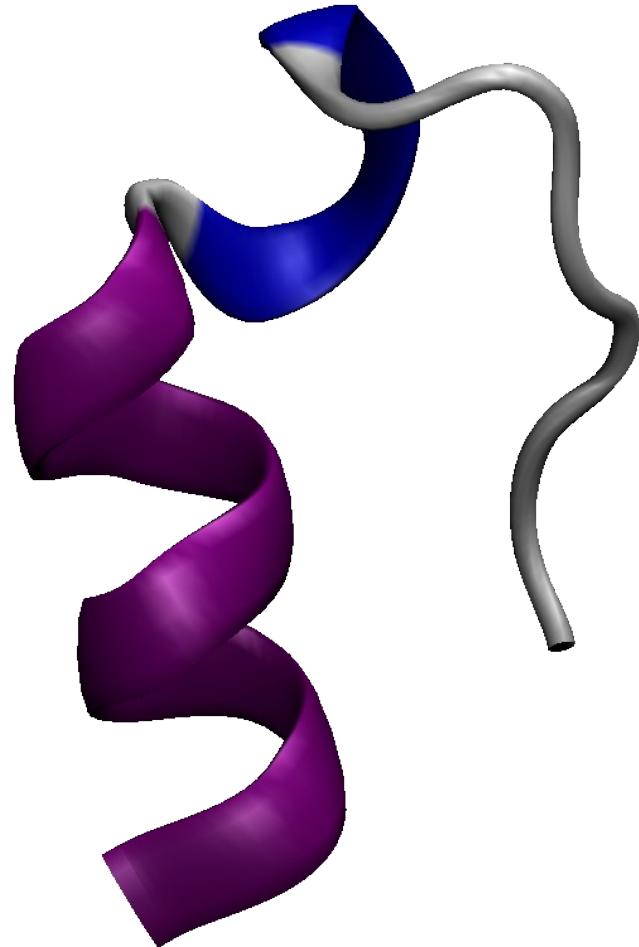
GROMACS in the Queue

- to wall clock time use: `mdrun -maxh xxx`
- `mdrun` creates checkpoint every 15min (`-cpt` option)
- to start from a checkpoint use `-cpi` option (smart option)
- chain jobs¹⁰

```
...script header
mdrun -maxh 36 -cpi state.cpt -append
[ $? -eq 0 ] && [ ! -f confout.gro ] && msub $0
```

¹⁰ `moab: msub -l depend=jobid xxx.log`

“Folding” the Trp-Cage Miniprotein



PDB ID: 1L2Y

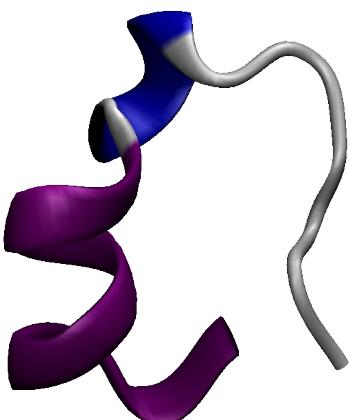
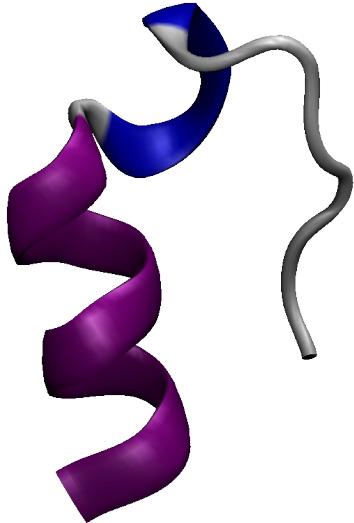
Summary of the Protocol

- Construct the needed topology files
- Minimize the folded conformation
- Simulate folded state in GBSA/OBC Implicit Solvent
- Construct extended protein chain (unfolded conformation)
- Perform high-to-low temperature annealing (attempt to fold the protein)
- Analysis of the results
 - Trajectory smoothing
 - RMSD to folded structure
 - Clustering
 - Dimensionality Reduction

Onufriev, A., Bashford, D., & Case, D. A. (2004). Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins*, 55(2), 383–394. doi:10.1002/prot.20033

Getting Started

- *Grab the tutorial files:*
 - cp /home/jphillips/tutorial.tgz .
 - scp darwin:/home/jphillips/tutorial.tgz .
- *Unpack the files:*
 - tar xzf tutorial.tgz
- *Change into the tutorial/work directory*



Prepare Topology

```
pdb2gmx -f ../pdb/1L2Y.01.pdb -o trp-cage.gro -p trp-cage.top -i  
trp-cage.itp -ignh
```

Prepare Box

```
editconf -f trp-cage.gro -o trp-cage.box.gro -box 5.5 5.5 5.5
```

Minimize Structure

```
grompp -f ../protocol/01-min.mdp -c trp-cage.box.gro -p trp-  
cage.top -o trp-cage.min.tpr  
mdrun -v -deffnm trp-cage.min
```

Run Equilibration

```
grompp -f ../protocol/02-gbsa-equilibration.mdp -c trp-  
cage.min.gro -p trp-cage.top -o trp-cage.equil.tpr -maxwarn 1  
mdrun -deffnm trp-cage.equil
```

Look at results

```
g_filter -s trp-cage.equil.tpr -f trp-cage.equil.xtc -all -ol trp-  
cage.equil.lp.xtc
```

Analyze RMSD

```
g_rms -s trp-cage.box.gro -f trp-cage.equil.xtc  
xmgrace rmsd.xvg
```

Grace: rmsd.xvg



File Edit Data Plot View Window

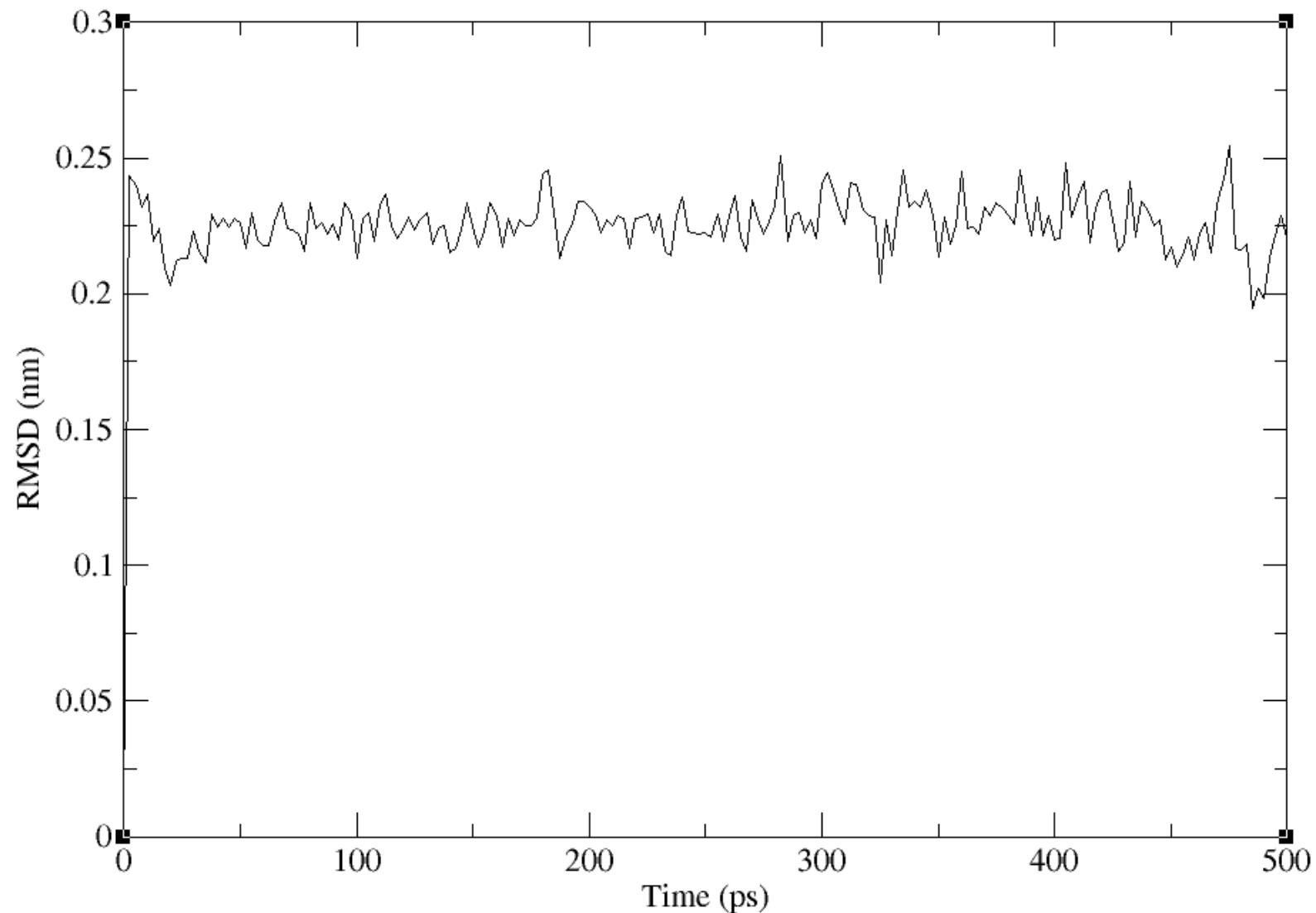
Help

GO: X, Y = [299.387, 0.248739]

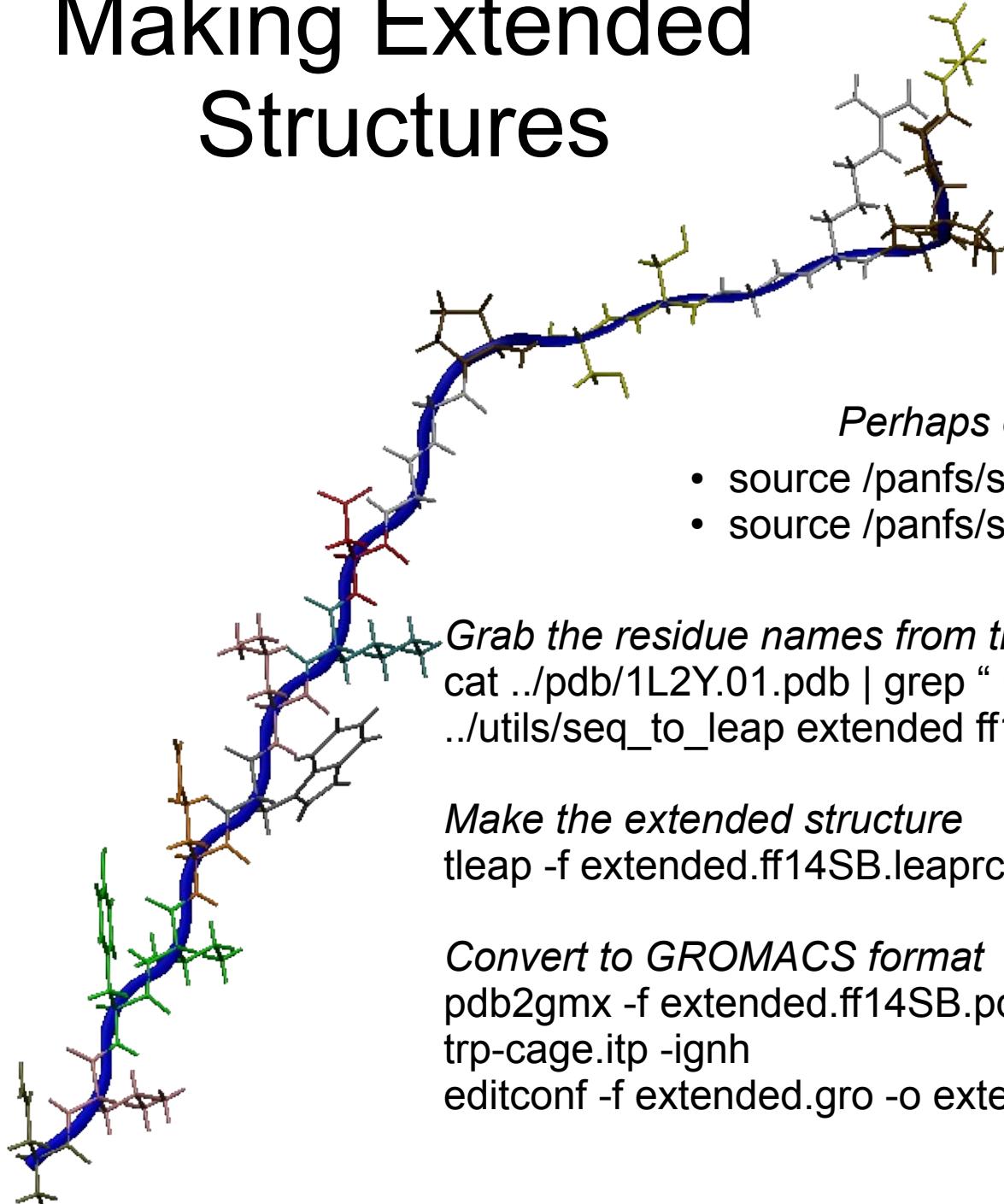
Draw
X A_S
Z z
← →
↓ ↑
AutoT
AutoO
ZX ZY
AX AY
PX Pu
Po Cy
SD:1
CW:0
Exit

RMSD

Backbone after lsq fit to Backbone



Making Extended Structures



- tar xjf AmberTools14.tar.bz2
- cd amber14
- export AMBERHOME=\${PWD}
- ./configure gnu
(Might as well install patches...)
- make -j 32 install
- export PATH=\${PATH}: \${PWD}/bin

Perhaps easier for this tutorial...

- source /panfs/scratch1/jphillips/apps/AMBER.bash
- source /panfs/scratch1/jphillips/apps/MDSCTK.bash

Grab the residue names from the PDB file

```
cat ..../pdb/1L2Y.01.pdb | grep " CA " | cut -c18-20 |  
..../utils/seq_to_leap extended ff14SB > extended.ff14SB.leaprc
```

Make the extended structure

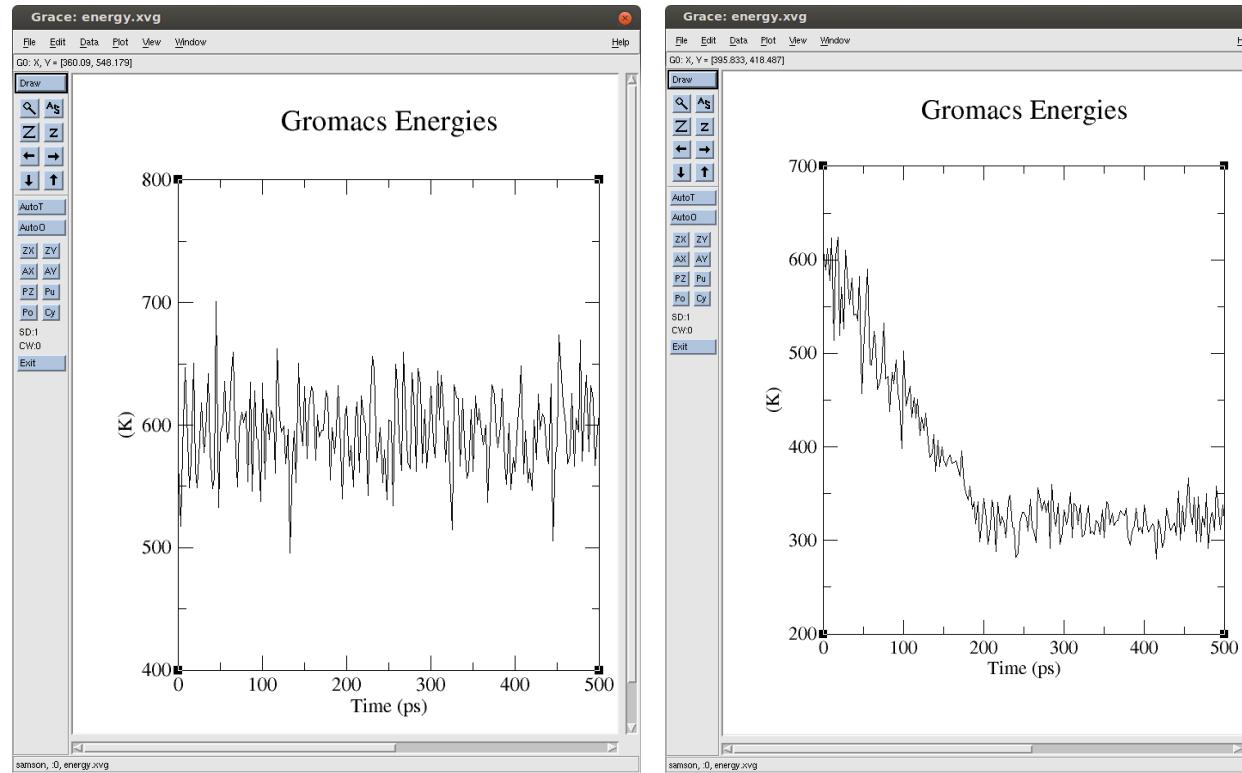
```
tleap -f extended.ff14SB.leaprc
```

Convert to GROMACS format

```
 pdb2gmx -f extended.ff14SB.pdb -o extended.gro -p trp-cage.top -i  
trp-cage.itp -ignh  
editconf -f extended.gro -o extended.box.gro -box 5.5 5.5 5.5
```

Minimize the system

```
grompp -f ../protocol/01-min.mdp -c extended.box.gro -p trp-cage.top -o extended.min.tpr  
mdrun -deffnm extended.min
```



Run Annealing

```
grompp -f ../protocol/03-gbsa-heated.mdp -c extended.min.gro -p trp-cage.top -o trp-  
cage.heated.tpr -maxwarn 1  
mdrun -deffnm trp-cage.heated
```

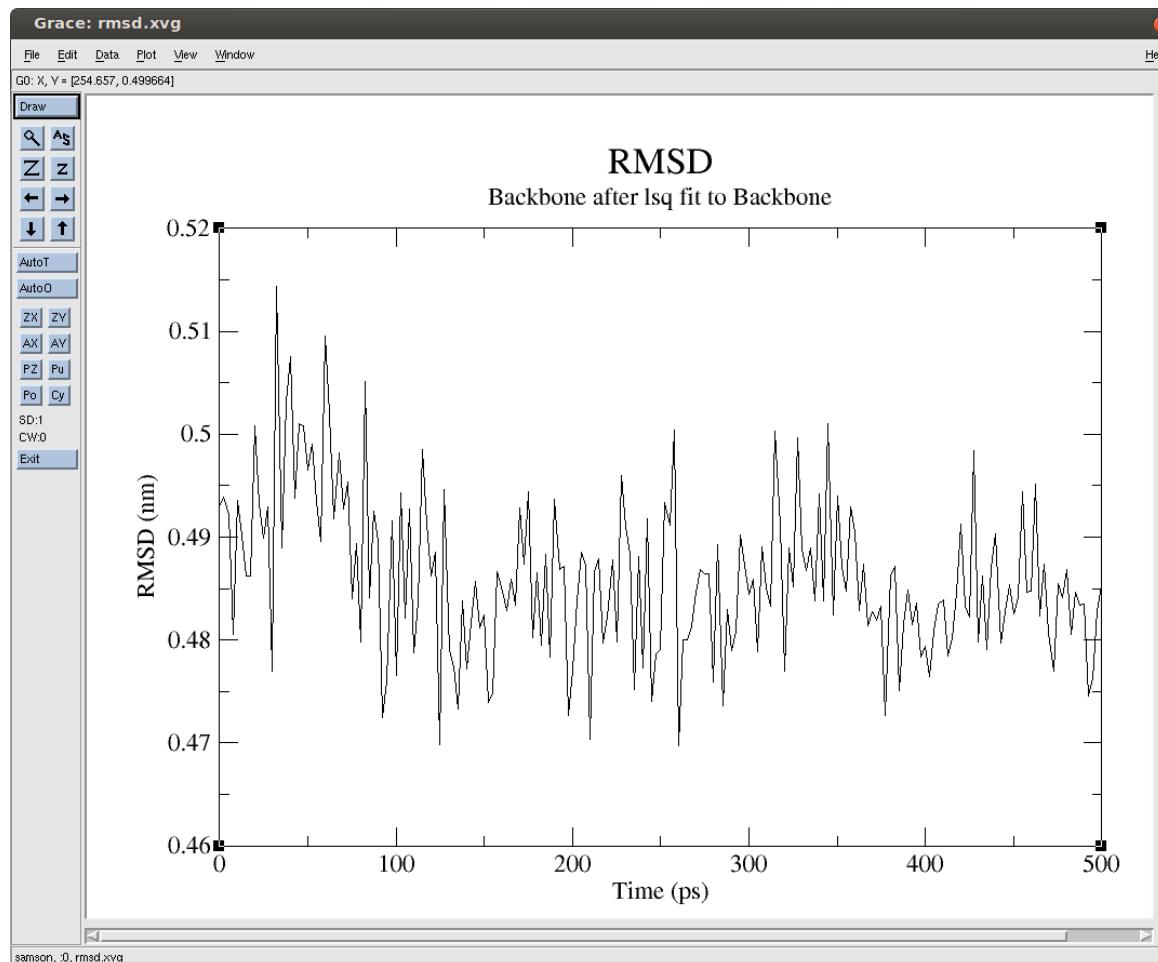
```
grompp -f ../protocol/04-gbsa-cooled.mdp -c trp-cage.heated.gro -p trp-cage.top -o  
trp-cage.cooled.tpr -maxwarn 1  
mdrun -deffnm trp-cage.cooled
```

Look at results

```
g_filter -s extended.cooled.tpr -f extended.cooled.xtc -all -ol  
extended.cooled.lp.xtc
```

Analyze RMSD

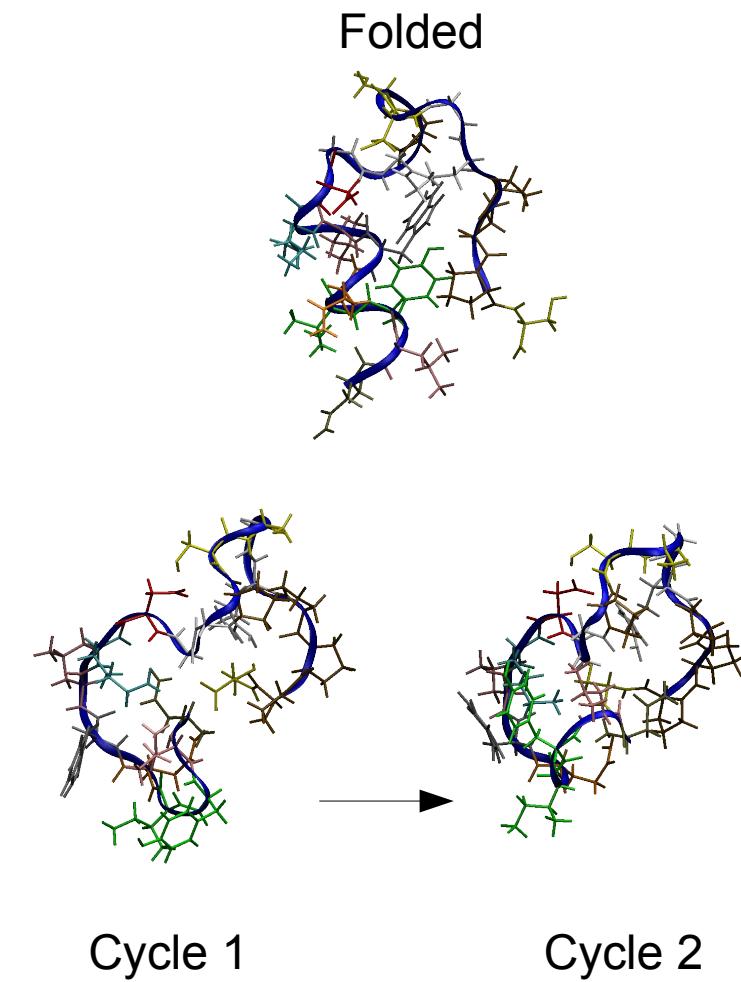
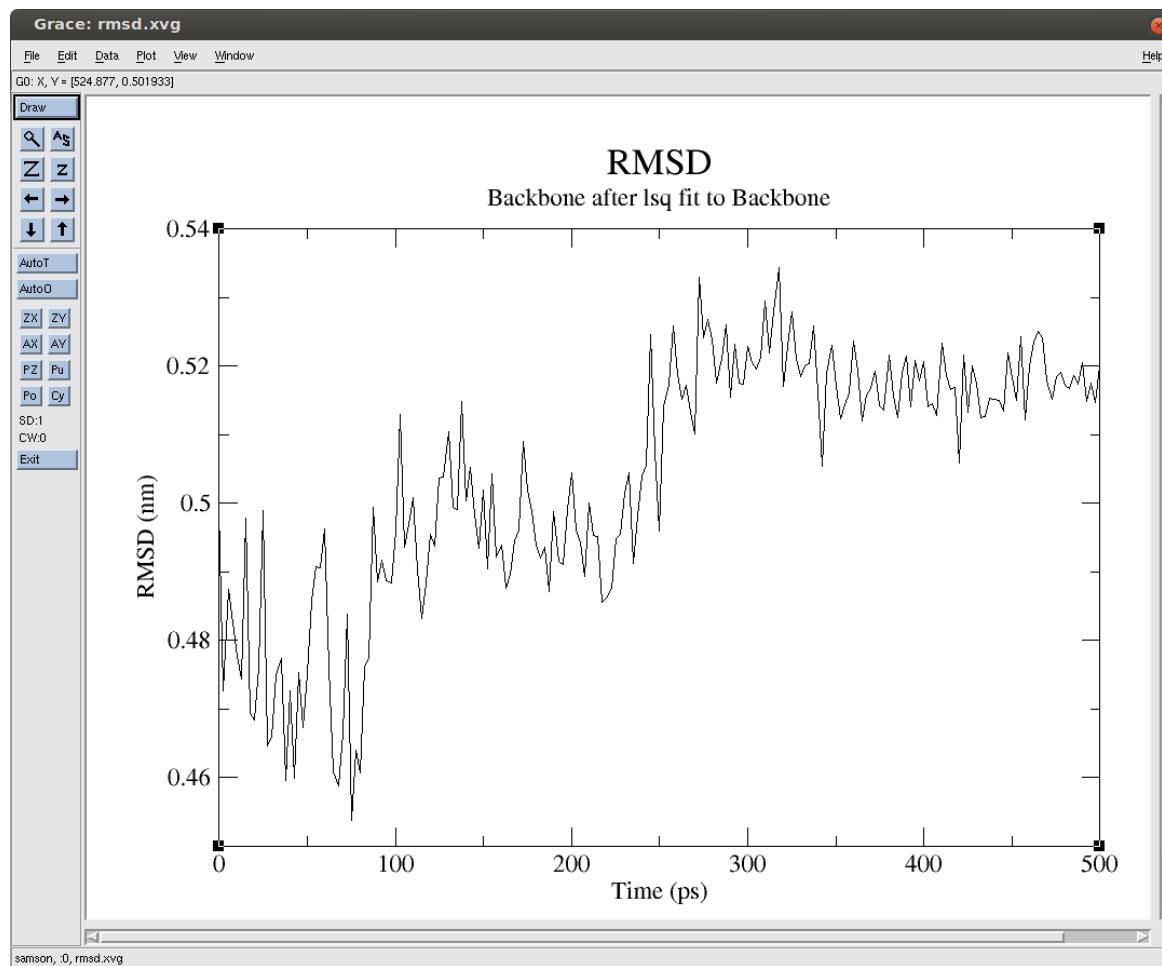
```
g_rms -s trp-cage.box.gro -f extended.cooled.xtc  
xmGrace rmsd.xvg
```



Re-Run Annealing

```
grompp -f ../protocol/05-gbsa-reheatedmdp -c extended.cooled.gro -p trp-cage.top  
-o extended.reheated.tpr -maxwarn 1  
mdrun -deffnm extended.reheated
```

```
grompp -f ../protocol/04-gbsa-cooled.mdp -c extended.reheated.gro -p trp-cage.top  
-o trp-cage.cooled2.mdp -maxwarn 1  
mdrun -deffnm trp-cage.cooled2
```



Spectral Clustering

Combine trajectories

```
trjcat -cat -o combined.xtc -f extended.heated.xtc extended.cooled.xtc  
extended.reheated.xtc extended.cooled2.xtc
```

Perform clustering

```
knn_rms -t 8 -p trp-cage.box.gro -r combined.xtc -k 256
```

```
make_sysparse -k 256
```

```
auto_decomp_sparse -k 20 -n 4
```

```
kmeans.r -k 4
```

```
Rscript -e 'write(sort(rep(seq(1,4),201)),ncolumns=1)'
```

```
mv data assignment.dat
```

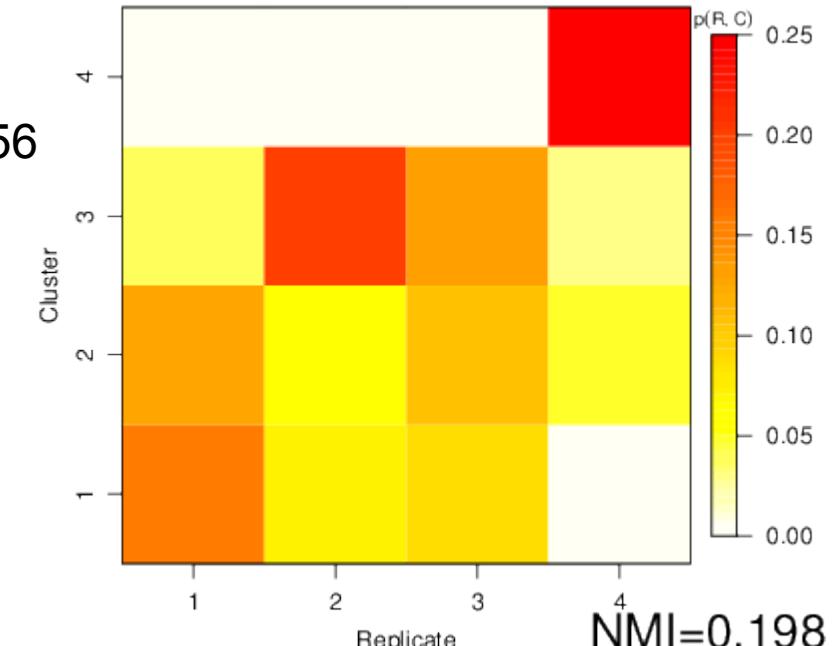
```
clustering_pdf.r
```

```
plot_pdf.r
```

```
evince pdf.eps
```

MDSCTK Compilation on Darwin

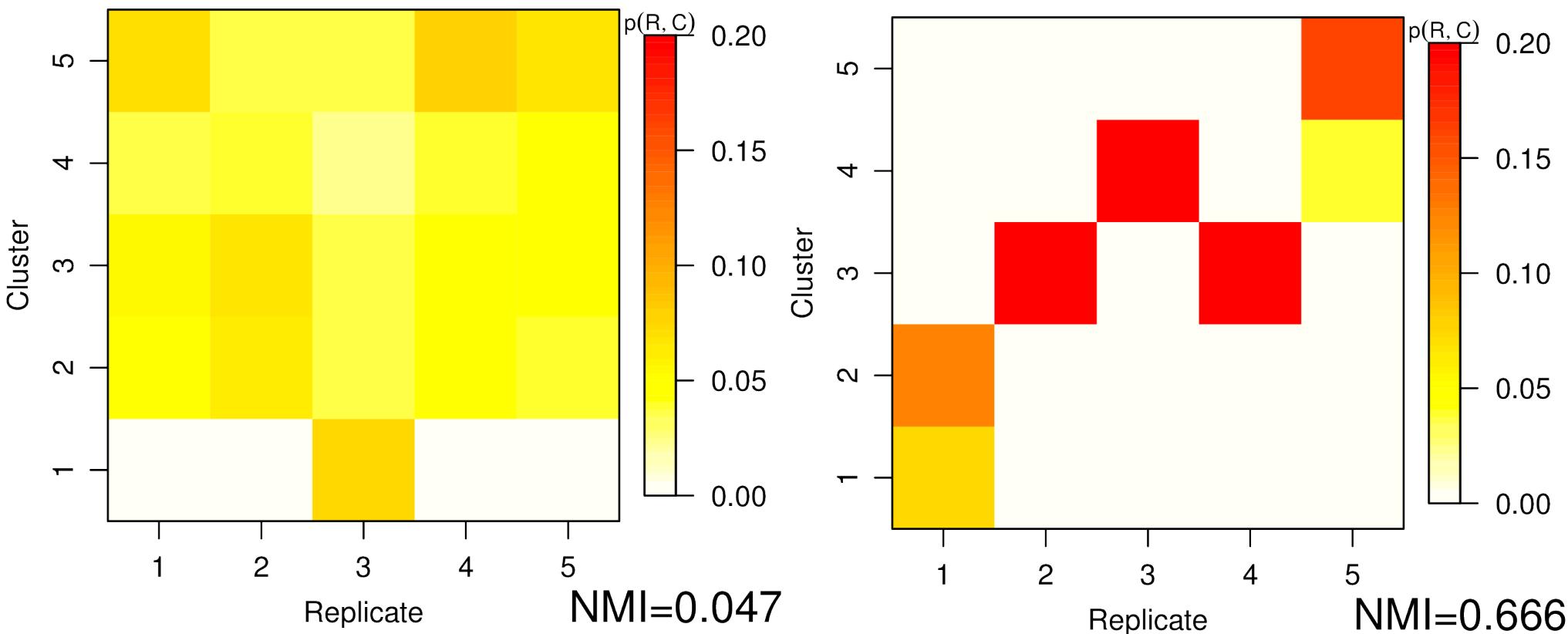
- git clone <https://github.com/douradopalmares/mdsctk.git>
- tar xzf mdsctk-1.2.2.tar.gz
- module load cmake
- module load gromacs
- cd mdsctk-1.2.2; cmake .
- make -j 12
- source MDSCTK.bash



Perhaps easier for this tutorial...

- source /panfs/scratch1/jphillips/apps/MDSCTK.bash

Results for FG-Nups



FG-Nup Motif	Spectral 3ns	Spectral 18ns	K-means 3ns	K-means 18ns
SxSG	0.138	0.047	0.147	0.051
AxAG	0.686	0.089	0.683	0.116
FxFG	0.841	0.666	0.841	0.521
GALG	0.864	1.000	0.855	0.526
GAFG	0.912	1.000	0.959	0.762
GLFG	0.839	1.000	0.949	1.000

Isomap

*Perform dimensionality reduction
(after knn_rms step, or after
spectral clustering)*

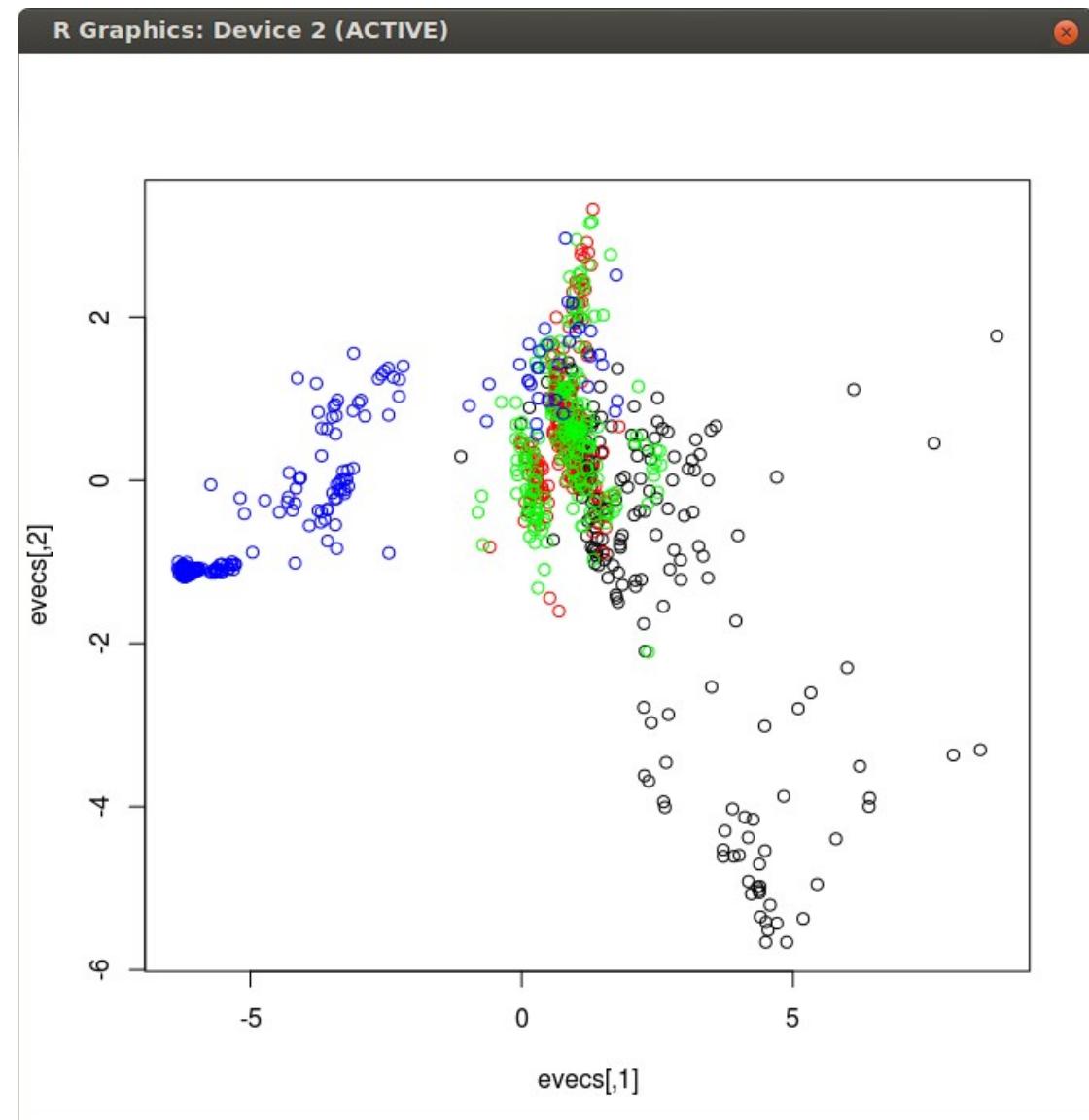
make_sysparse -k 256 -n 32

dijkstra

decomp_dense

R

```
evecs ←  
matrix(scan("eigenvectors.dat"),ncol  
=804)  
plot(evecs,col=c("black","red","green","blue")[sort(rep(seq(1,4),201))])
```



Replica Exchange MD

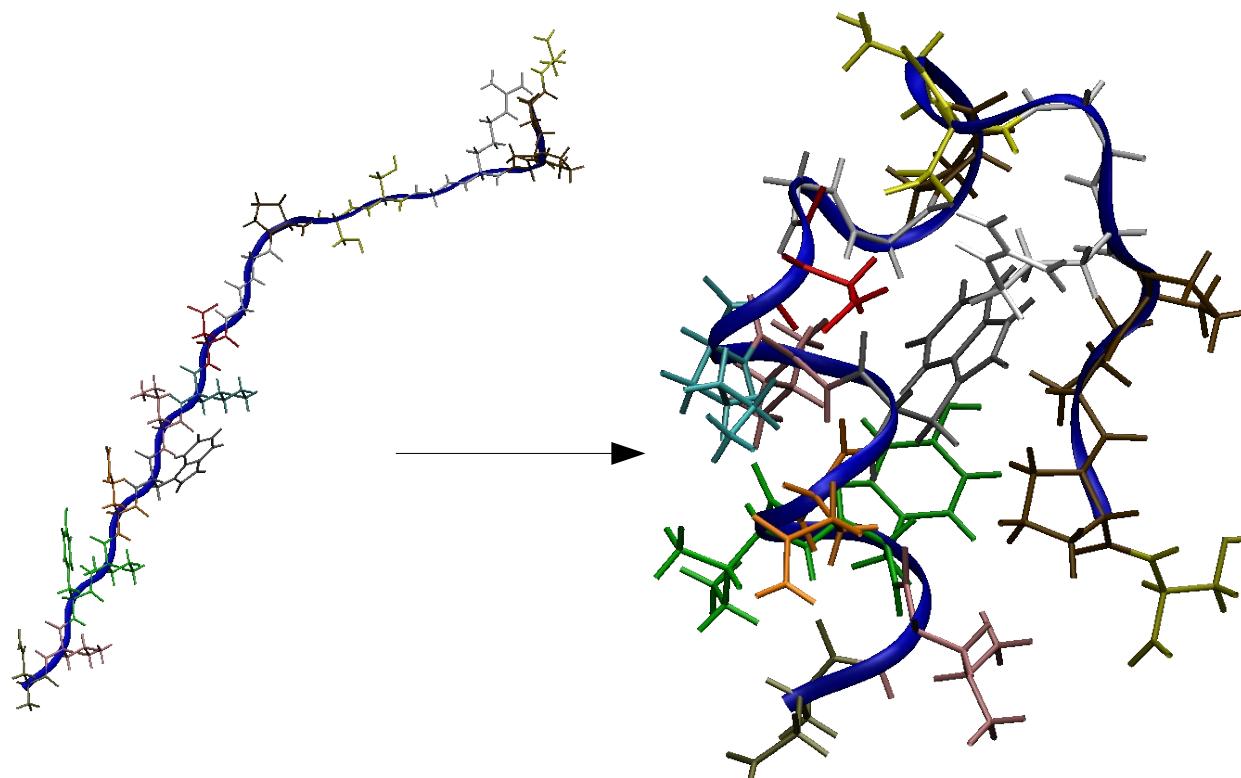
Setup Temperature Ladder

```
grompp -f ../protocol/06-gbsa-replex-#.mdp -c extended.min.gro -p trp-cage.top -o extended.replex-#.tpr -maxwarn 1
```

Performed for each MPD file (temperature)

Run it using MPI

```
mpiexec -n 8 mdrun_mpi -s extended.replex-.tpr -multi 8 -replex 50 -deffnm  
extended.replex-
```



Crystalline Lysozyme

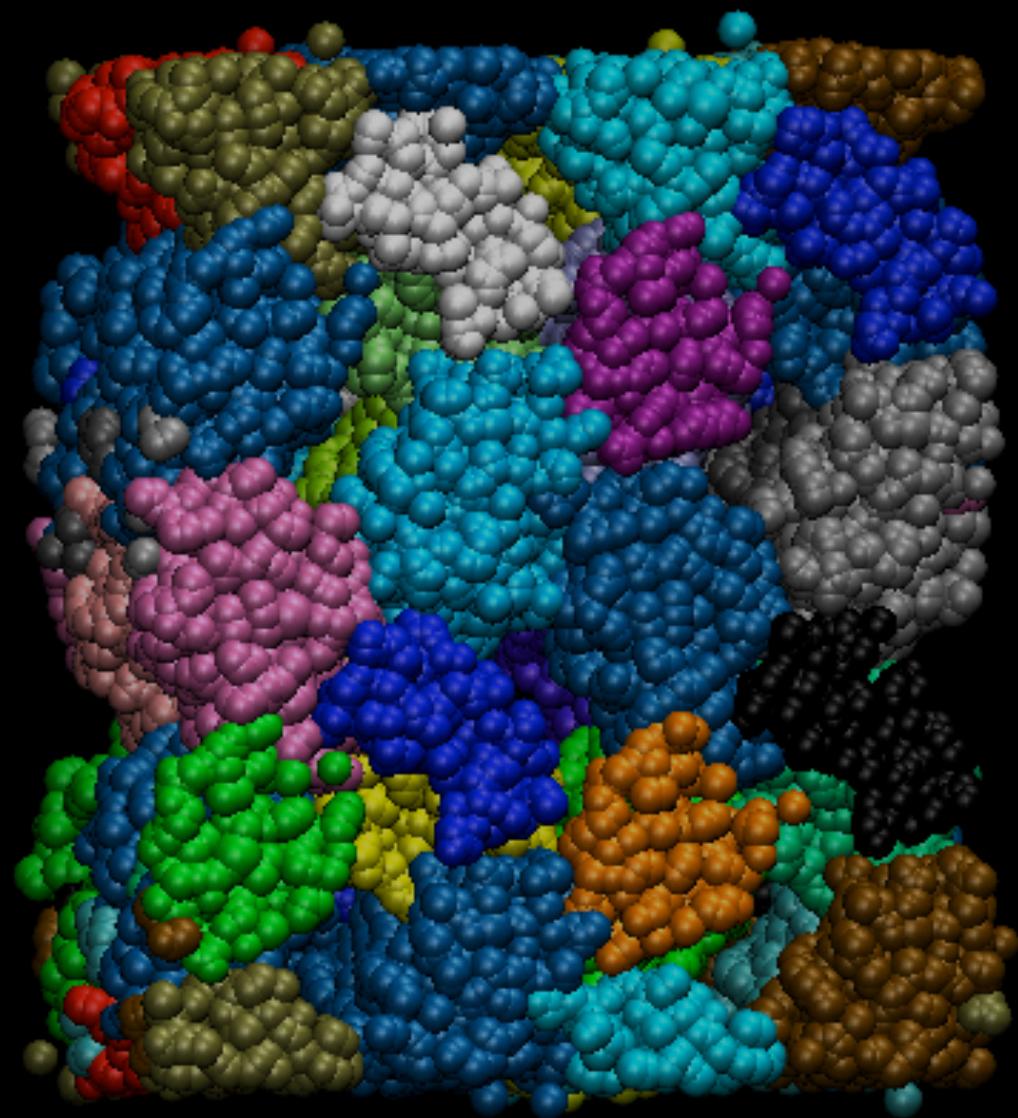
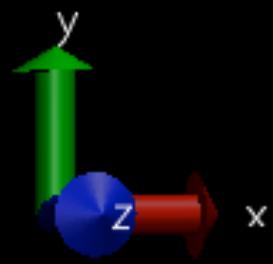
- Published
 - 1 ns crystalline lysozyme simulation using CHARMM
 - Hery, Genest, and Smith. J Mol Biol 279 (1998) 303
- This example
 - 1 ns using 1AKI
 - GROMACS 4.6.5
 - AMBER99SB-ILDN (good choice according to Case lab)

Crystalline lysozyme

- Use AmberTools to generate model of 2x2x2 unit cells
- Preparation similar to solvated lysozyme example
 - editconf_d -f 1AKI_prop_processed.gro -o 1AKI_prop_newbox.gro -c -box 11.8124 13.6902 6.1034

Visualization in VMD

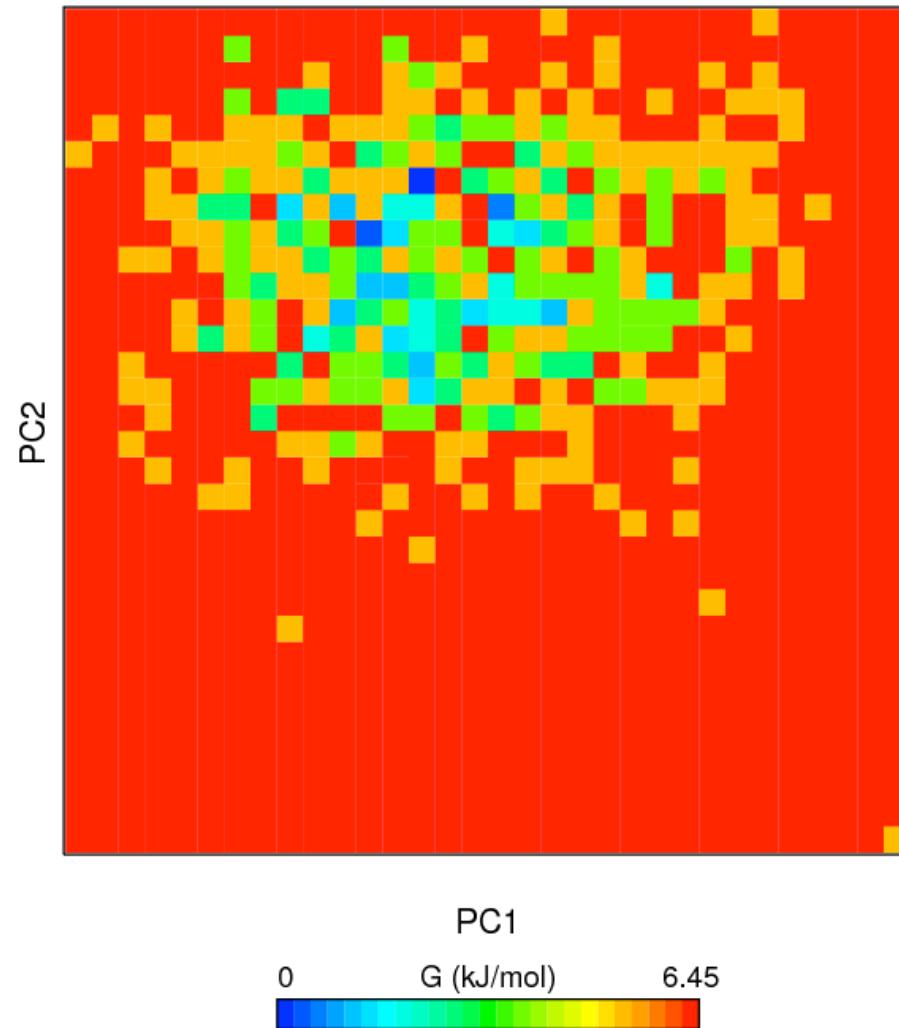
- `trjconv_d -f md_npt_1ns.xtc -o
md_npt_1ns_nojump.xtc -s md_npt_1ns.tpr -pbc
nojump`
- `trjconv_d -f md_npt_1ns_nojump.xtc -o
md_npt_1ns_nojump_fit.xtc -s md_npt_1ns.tpr -
fit translation`
- `trjconv_d -f md_npt_1ns_nojump_fit.xtc -s
md_npt_1ns.tpr -o
md_npt_1ns_nojump_fit_20ps.pdb -dt 20`
- Read file `md_npt_1ns_nojump_fit_20ps.pdb` into
VMD



Visualization of Energy Landscape

- make_ndx_d -f md_npt_1ns.tpr -o A-D_CA.ndx
- trjconv_d -f md_npt_1ns_nojump_fit.xtc -s md_npt_1ns.tpr -o md_npt_1ns_A-D_CA.xtc -n A-D_CA.ndx
- tpbconv_d -s md_npt_1ns.tpr -n A-D_CA.ndx -o md_npt_1ns_A-D_CA.tpr
- g_covar_d -f md_npt_1ns_A-D_CA.xtc -s md_npt_1ns_A-D_CA.tpr
- g_anaeig_d -2d -v eigenvec.trr -eig eigenval.xvg -f md_npt_1ns_A-D_CA.xtc -s md_npt_1ns_A-D_CA.tpr -first 7 -last 8
- g_sham_d -f 2dproj.xvg -ls gibbs.xpm –notime
- xpm2ps_d -f gibbs.xpm -o gibbs.eps -rainbow red

Gibbs Energy Landscape



Conclusion

- GROMACS is great!
- GROMACS is fast!
- GROMACS is free!
- There is a manual out there! (RTFM)
- Join gmx-users list
- Join gromacs@lanl.gov list (or tell me)

The End

Thank you for your attention !

